



# WEB APPLICATION FOR SUPPORTING THE ENTRANCE EXAM ORGANIZATION

Ana Petrović<sup>1</sup>,  
Violeta Tomašević<sup>2</sup>

<sup>1</sup>VIŠSS Tehnikum Taurunum,  
Nade Dimić 4,  
Zemun, Serbia  
<sup>2</sup>Singidunum University,  
Danijelova 32,  
Belgrade, Serbia

## Abstract:

A web application for supporting the organization of entrance exams is described in this paper. It is intended for university examination purposes. From a functional point of view, the application consists of three parts: the registration of candidates, planning the seating arrangements of candidates, and the evaluation of exam results. This client-server application is implemented within the NetBeans visual development environment using the open-source XAMPP package. The presentation level is realized with standard HTML and CSS web technologies, the business logic is written in the PHP programming language, while data level is implemented using the MariaDB database management system.

## Keywords:

entrance exam; organization; ranking list; web application.

## 1. INTRODUCTION

During the last decade, contemporary society has experienced an extremely rapid technological progress primarily due to connecting people over the world by the global computer network – the Internet. Significant changes have occurred in all areas of the society, including the area of education [1]. In the higher education domain, new web-based learning methods, new ways of taking exams and assessing students, as well as distance learning facilities have emerged. These advances are primarily the result of an intensive use of various education software solutions. In the education field, the web-based approach is primarily focused on students, then teachers, and finally on administrative staff.

Although education software tools cover a wide range of functionalities, from simpler tools with specific purposes [2] to complex, universal applications [3], few of them are dedicated to school administration activities. Administrative staff in higher education mostly uses an information system to efficiently perform everyday tasks. However, they are constantly faced with the lack of applications intended for some tasks that are important for the institution, but rarely performed (for example, entrance exam organization, student internship planning, alumni association events planning, etc.).

The SEEO (Support for Entrance Exam Organization) application is intended only for administrative staff. Its aim is to provide an assistance

## Correspondence:

Ana Petrović

## e-mail:

ana.petrovic.152@singimail.rs



in organizing entrance exams at a higher education institution. The organizers of entrance exams can encounter various problems. For example, in case of a large number of candidates, searching for the information about the specific applicant can be hard to perform. In this case, it often happens that the documentation about the candidate is incomplete. After the exam seating arrangement is made, each change in the availability of classrooms implies a serious time-consuming problem. Since candidates always expect final results to be released as soon as possible, errors in the ranking lists are not rare, which also causes some inconvenience and may require additional time for correction. The SEEO application organizes and keeps all the information about the candidates who applied for the entrance exam in a systematical manner. Also, in a simple way it creates the exam seating arrangement and updates it if needed. After the exam is over, the application automatically generates the ranking list of candidates according to their achieved results.

In the open literature, applications dedicated to entrance exams are rarely found. Existing applications are mainly intended for students as a help in preparing the entrance exam. In our environment, an application offering support for the entrance exam at the School of Electrical Engineering, University of Belgrade, attracts one's attention [4]. It is a part of their information system and enables the organization of entrance exams for more schools with different profiles and specific grading rules at the same time. The exams can take place several days at arbitrary locations, which indicates that this is a comprehensive and versatile software tool.

This paper consists of four parts. After an introduction, the functionalities of the SEEO application are described. Details about the implementation of this application are presented in section III. Finally, the conclusion briefly summarizes the features of the application and gives some suggestions for future improvements.

## 2. FUNCTIONAL DESCRIPTION

Only a registered user can start working with the SEEO application by logging in with his username and password. Registration is allowed only for the staff involved in the entrance exam organization.

Regarding its functionality, the SEEO application consists of three parts:

- ◆ Registration of candidates
- ◆ Planning the seating arrangements of the candidates
- ◆ Evaluation of the results

### *Registration of candidates*

Application of candidates for entrance exam on a higher-education institution assumes collecting some necessary information about a candidate: basic personal data, the record about success in high school, and his wishes and choices regarding the department, model of studying, etc.

Basic personal data of a candidate include: first name and family name, name of a parent, personal identification number (PID), phone number and an e-mail address.

When accepting the personal identification number which consists of 13 digits, the system performs a check of its validity. Each digit is associated with a letter of the alphabet (first digit with a, and so on), and then the control digit is calculated as:

$$11 - ((7(a+g) + 6(b+h) + 5(c+i) + 4(d+j) + 3(e+k) + 2(f+l)) \% 11)$$

If the control digit matches the last digit of PID, it is considered valid. Otherwise, it is required to enter PID once again.

Candidates's achievement during high school education (GPA, as well individual grades for specific courses relevant for the higher-education institution for which the candidate is applying) are compiled in an aggregate score according to the official rules before it is entered into the application.

During application process, a candidate also expresses his preference for the desired department and the model of studying. The department is chosen from the offered list of available ones. Within the department, the candidate can choose a specific module, if available. The candidate also chooses between two models of studying: state-funded or self-financed.

After the application is completed, the candidate is automatically associated with a unique identifier in the form of:

*xyznnn*

where *x* is – a department ID, *y* – a module ID, *z* – the model of studying, and *nnn* – the ordinal number of application on a department and module.

The stored data about a candidate (excluding his identifier) can be modified if necessary. Also, a candidate can be removed from the entrance exam registration list.

The SEEO application provides a spreadsheet of all registered candidates sorted by their identifiers. It also enables to search for the candidates using appropriate parameters (name, PID, etc.).



### Planning the seating arrangements

In order to generate the seating arrangement for the entrance exam by the SEEO application, it is necessary to define the set of classrooms for this purpose. The set of classrooms can be modified only before the assigning of seats to candidates has started.

For each new classroom two items are needed: its label number and its capacity (the number of available seats). While arranging the seating plan, the application automatically updates the numbers of remaining available seats in classrooms.

Planning the seating arrangements for an entrance exam is an iterative process on the round-robin basis. Registered candidates are picked from the list sorted by their identifiers and iteratively allocated to a classroom in the increasing order of classroom label numbers (the first candidate goes to the first classroom, etc.). When the first round is over, and each classroom obtained one candidate, the next candidate from the list goes to first classroom again. When a classroom is filled up (the capacities of the classrooms are generally different), it is not considered any more in the next round.

During the process of application, the identifiers are assigned to the candidates sequentially as they apply. The round robin seating assignment avoids the situation where two candidates who know each other (they applied together) are meant to be seated in the same classroom, next to each other. In this way, it contributes to the regularity of an entrance exam.

After the seating arrangement is completed, an appropriate list of the candidates for each classroom is generated. Additional statistics on availability of classrooms and numbers of free seats can be obtained.

Candidates who registered later are assigned to classrooms in the same way, starting from the current state of classrooms' availability.

### Evaluation of the results

After the entrance exam is over and the work of the candidates is graded, the score achieved for each candidate is entered into the application. It is accumulated with the points assigned according to the success in high-school education and ranking lists sorted by the total score are generated. A separate ranking list is generated per each department and module (Fig.1).

On each ranking list two borderlines are clearly highlighted. The first borderline on the list is set before the

first candidate with a score lower than 51% points. The candidates above this limit are enrolled as state-financed students. The second borderline is set before the first candidate with a score lower than 30% points. The candidates between two borderlines are also allowed to enroll as self-financed students. The candidates below the second borderline are not eligible to enroll.

Redni broj	Šifra kandidata	Ime i prezime	Ukupno bodova
1	112005	Andrija Poletanović	82.56
2	111001	Zoran Rendulić	80.00
3	111002	Nemanja Nedović	55.00
***** Granica 1: Upis studenata na teret budzeta *****			
4	111003	Marjana Simović	50.00
***** Granica 2: Upis samofinansirajucih studenata *****			
5	111004	Jovana Visković	30.00

Fig. 1. An example of the ranking list.

## 3. IMPLEMENTATION

The SEEO application is implemented using the model of three-layer client-server architecture, as it is illustrated in Fig. 2.

The first layer represents the presentation level which realizes the user-application interface. It is implemented in a standard way using HTML technology [5]. The web browser on a client computer interprets HTML code and results are displayed to the user. Uniform format of web pages within the application is obtained using CSS technology [6].

The second layer implements the business logic of the application. This layer is realized in PHP programming language [7,8]. The PHP code doesn't require compilation since it is interpreted on the web server. As it is primarily intended for web programming, the PHP code can be embedded into HTML elements, which facilitates the development process and makes it more efficient. In addition, PHP offers various libraries to support different databases. These features make it very convenient for linking and synchronizing the presentation level and data level.

The third layer enables storing of data relevant for the application and preserves their security and integrity. The data layer is based on the database management system (DBMS). This application employs MariaDB DBMS. It has been chosen as it is a free-licensed open code software which took the place of MySQL after its commercialization. Until recently, MySQL was a standard for use in



a small-scale and middle scale application, so it can be expected that MariaDB can take its place in the future because of their compatibility and certain enhancements (faster execution, multithreading in handling queries, etc.)

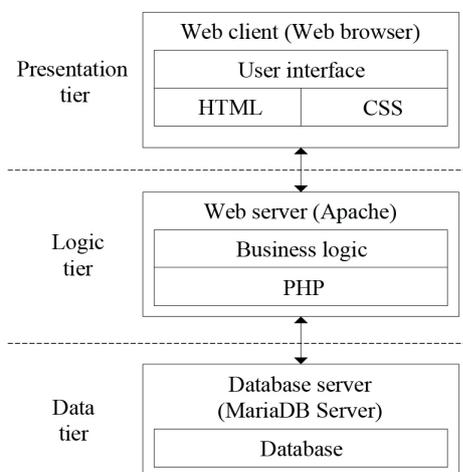


Fig. 2. Application architecture.

General functioning of the application will be illustrated using the example of registration of a new candidate. When a user clicks the *Add candidate* button, the web browser on the client computer interprets a corresponding HTML code and opens a page for entering data about the candidate. After data are entered into appropriate fields the user clicks the Save field. The web browser sends to the server a HTTP request for storing data into the database. The web server reads the database with the PHP code which implements the update of the database. The contents of the database are forwarded by the web server to the PHP interpreter for handling. Then, the interpreter performs the syntax analysis of the script. Since in this case the script contains the instructions for database access, PHP establishes the connection with the database server and sends an appropriate query to it. The database server executes the query by storing the received data about the candidate into the corresponding table in the database and sends the result to the PHP interpreter. This component completes the script handling. It also assumes the formatting of the outcome of the query into the HTML code which is then sent to the web server. In this situation, the outcome of the query is the display of data about the candidate in the list of all candidates already registered in the system. The web server forwards the obtained HTML code to the web browser which presents it to the user.

The entire implementation of the SEEO application is based on *open source* software. NetBeans IDE 8.1 [9]

was chosen for the development environment. This environment is widely used nowadays and offers a very comfortable editor which, among other, supports the programming languages used in this case, HTML and PHP. In addition, it is intended for work on different platforms (Windows, Linux, Mac).

During implementation the XAMPP Control Panel v 3.2.2 [10] software package is exploited. This package enables the testing and development of the application on a local computer and consists of independently created programs which operate on different platforms. In building our application, we used the following components of the XAMPP package: Apache web server, MariaDB DBMS [11] and the PHP interpreter. Yet another tool from the package was actively used – the phpMyAdmin application [12]. It provides graphical interface for convenient management of database.

In the design of the SEEO application the modular approach is followed. The implementation of system functionalities is organized into three basic modules: *Candidates*, *Classrooms*, and *RankingLists*. The *Candidates* module encompasses the functions for entering data about candidates and their choices for intended studying. The *Classrooms* module consists of the functions which enable the setting of available classrooms, their seating capacities and seating arrangements of registered candidates by classrooms. The *RankingLists* module covers the functions for entrance of data about the results achieved on the entrance exam and creates the ranking lists for different departments/modules. There is also an additional module with utility functions for application and registration of the candidates, display of information in page headers, transition between displays, termination of the application, etc.

#### 4. CONCLUSION

In the last decade, we evidenced a rapid expansion of high-level education, prevalently because of emergence of private institutions. Besides traditional state universities, numerous private high-level education institutions such as universities and colleges have appeared. Because of a wide variety of study programs and models of studying, the number of those who finished high schools and who are interested in continuing their education has increased. The government also encourages this trend in an effort to increase the percentage of population with a high-level education degree. The legal obligation of all high-level education institutions is to organize entrance exams in the process of selection of prospective students. In many



occasions the number of applied candidates is very high (sometimes several times higher than the number of enrollees), the organization of the entrance exam is a serious task. First of all, the regularity of the exam (preventing cheating, unallowed help, etc.) must be preserved which increases the number of classrooms and required staff. Also, before ranking lists are created, a fast, efficient and reliable handling of exam results must be provided.

All these requirements are practically impossible to meet without an appropriate software tool. However, there is lack of applications that could assist in the organization of entrance exams in our environment. This was just motivation for development of the SEEO application.

From a functional point of view, the SEEO application provides different conveniences in creating the consistent and complete data set for candidates, in organizing an entrance exam and forming well-organized and expressive ranking lists. The application also supports the configuration of available classroom pool for conducting an entrance exam. The change of classrooms, allocation of new ones and deallocation of existing ones are easy to manage. The main feature is planning the seating arrangement according to available classrooms on the round-robin basis. The evaluation of achieved results is itemized by departments and modules with highlighted borders for different means of student financing.

From the aspect of implementation, the SEEO application represents a standard web application whose development employed the usual *open source* web technologies (HTML and CSS on the client side and PHP and MariaDB on the server side). The advantage of web-based implementation also include the fact that the classroom attendant has on-line data available during the exam and can perform checks, if necessary.

The SEEO application represents an advanced version of the application previously developed for administrative staff of High Engineering School of Applied Studies Tehnikum Taurunum, Zemun, Serbia. It is expected that future experience will give the estimation of its usefulness. A more detailed consideration of organizational needs of the entrance exam should lead to additional functional requirements and further improvement of the application. After that, the SEEO application can be offered to other universities and colleges as support in the organization of their entrance exams.

In order to further upgrade the SEEO application we intend to thoroughly consider its security aspects (since it is a web application), as well as enhance its functionality. Adding new functions should be a permanent process until it becomes a comprehensive system which will provide

not only its main entrance exam related functionality, but also support for numerous activities specific to higher-level education institutions.

There is an imminent need to upgrade the application by adding a feature that will be able to manage the staff on an exam and appoint attendants across the classrooms. Moreover, it should increase flexibility in terms of fulfilling candidates' wishes. If a candidate doesn't achieve the score required for the enrollment in the desired department but has sufficient score for another, this option should be offered to him. The possibility of archiving entrance exam results should also represent a significant advantage. It would allow to follow and compare the results of an entrance exam for the higher education institution over a longer period of time. In this way, some relevant information can be obtained which could have an impact on the structure and improvement of study programs.

## REFERENCES

- [1] T. Lynch and C. Lynch, „Web Based Education“, The Innovation Journal: The Public Sector Innovation Journal, vol.8, no.4, 2008.
- [2] M. Pilja and V. Tomašević, “Evidencija – web aplikacija za praćenje nastavnih aktivnosti”, Telekomunikacioni forum TELFOR, str. 1500–1503, 2012. (in Serbian)
- [3] J. Goddard and P. Chatterton, „Space, place and 2012 the virtual university: the virtual university is the university-made concrete“, Conf. Virtual University? Educational Environment of the Future, Portland press, vol.79, 2003.
- [4] G. Ružić and S. Turajlić, „Informacioni sistem za organizaciju prijemnog ispita na Elektrotehničkom fakultetu u Beogradu“, ETRAN, 2008. (in Serbian)
- [5] HTML Tutorial, Refsnes Data 2016. Available: <http://www.w3schools.com/html/>
- [6] CSS Tutorial, Refsnes Data, 2016. Available: <http://www.w3schools.com/css/default.asp>
- [7] PHP Tutorial, Refsnes Data, 2016. Available: <http://www.w3schools.com/php/>
- [8] L. Welling and L. Thomson, PHP and MySQL: Web Development, ISBN 10:0-672-32916-6, USA, 2009.
- [9] NetBeans IDE - Overview, Oracle, 2016. Available: <https://netbeans.org/>
- [10] XAMPP Installers and Downloads for Apache Friends, Apache friends, 2016. Available: <https://www.apachefriends.org/index.html>



[11] Learn - MariaDB.org, MariaDB Foundation, 2016.  
Available: <https://mariadb.org/learn/>

[12] phpMyAdmin, phpMyAdmin contributors, 2003-2016. Available: <https://www.phpmyadmin.net/>